



DomainTools Iris Detect API

Generated on April 21, 2026

Contents

1	Sample Response	3
2	Iris API Authentication	4
2.1	Overview	4
2.2	Protecting API Credentials	4
2.3	Open Key Authentication	4
2.4	API Key (Header) Authentication	4
2.4.1	Required API Key Authentication Parameters	4
2.4.2	Example Request with API Key Authentication	5
2.5	HMAC Signed Authentication	5
2.5.1	HMAC-Signed Authentication URI/Parameters	5
2.5.2	Creating a HMAC-signed authentication request	5
2.5.3	Security Note	6
3	Iris Rate Limits	7
3.1	Check your rate limits	7
3.2	Free test queries	7
3.2.1	Endpoints with free test queries	7
3.3	Iris Investigate rate limits and quota consumption	7
3.3.1	Duplicate query policies	7
3.3.2	Activities that consume quota	8
3.4	Iris Enrich rate limits	8
3.5	Iris Detect rate limits	9
3.5.1	Domain endpoints rate limits	9
3.5.2	Development and testing	9
3.5.3	Other endpoints	9
4	Iris API Error Codes	10
4.1	Status Codes	10
4.2	Error Response Format	10
4.3	Common Error Scenarios	11
4.3.1	Authentication Errors (401)	11
4.3.2	Authorization Errors (403)	11

4.3.3	Not Found Errors (404)	11
4.3.4	Rate Limiting (503)	11
4.4	Best Practices	11
4.5	Iris-Specific Notes	12
4.5.1	Partial Content (206)	12
4.5.2	Detect Endpoints	12
4.5.3	Validation Errors (422)	12
5	Iris Detect API guide	14
5.1	Overview	14
5.2	About Iris Detect APIs	14
5.2.1	Functionality supported via APIs	14
5.2.2	Not supported by API	14
5.3	Getting started	14
5.4	Authentication	14
5.5	Authorization and permissions	15
5.5.1	Permission types	15
5.5.2	Required permissions for API actions	15
5.6	API rate limits	15
5.6.1	Endpoints with special rate limits	15
5.6.2	Bypassing rate limits during integration work	15
5.7	Finding and resetting the API key	16
5.8	API endpoints	16
5.8.1	Monitor list	16
5.8.2	Create monitor	17
5.8.3	Update monitor	19
5.8.4	Delete monitor	20
5.8.5	Domains	21
5.8.6	Ignored domains	25
5.8.7	Add and remove from watchlist	27
5.8.8	Escalate	28
5.9	Working with screenshots	30
5.10	Appendix: API response examples	30
5.10.1	Monitor list example	30
5.10.2	Domain list example	30
5.10.3	Add domain to be watched example	32
5.10.4	Escalate a domain for blocking example	33
6	Working with screenshots in Iris APIs	34
6.1	Retrieve the screenshot URL	34
6.1.1	Example queries for screenshot URL retrieval	34
6.1.2	Query parameters for screenshot URL retrieval	34
6.1.3	Response fields for screenshot URL retrieval	34
6.1.4	Response example for screenshot URL retrieval	35
6.2	Retrieve and optionally resize the screenshot image via its URL	35
6.2.1	Query parameters for screenshot image file	35
6.2.2	Example query for screenshot image file	36
6.3	Interpreting screenshot metadata	36
6.3.1	Initial capture	36
6.3.2	Gather attempt was duplicate of initial screenshot	36
6.3.3	Recapture failed after previous successful capture	37
6.3.4	Duplicate capture followed by failed attempt	37

The Account Information API provides a quick and easy way to get a snapshot of API product usage for an account. Usage is broken down by day and by month.

```
https://api.domaintools.com/v1/account/
```

1 Sample Response

```
{
  "response": {
    "account": {
      "api_username": "domaintools-api-account",
      "active": true
    },
    "products": [
      {
        "id": "domain-profile",
        "per_month_limit": "100000",
        "per_minute_limit": "120",
        "absolute_limit": "1000",
        "usage": {
          "today": 5,
          "month": 152
        },
        "expiration_date": "2020-01-01"
      }
    ]
  }
}
```

2 Iris API Authentication

2.1 Overview

Most requests sent to the DomainTools API require authentication. We support two authentication schemes with different levels of security.

We recommend the following practices for authentication:

- Authentication with HTTPS (HTTP over SSL) instead of HTTP.
- Using HMAC signed queries or Header Authentication instead of Open Key Authentication.
- Using SHA-256 HMAC for the signed queries instead of MD5 or SHA-1.

2.2 Protecting API Credentials

Your account will be charged for all queries authenticated with your username and key, even if you later determine the requests were fraudulent or its use unauthorized. We recommend the following steps to protect your API credentials:

- For support requests, do not send the full API key to DomainTools support.
- Use a HMAC-signed approach to requests. This ensures that the user and token are not sent as part of the URL.
- When required, reset the API key.
- Do not place them into a public repository, such as a git repository.

2.3 Open Key Authentication

This is the easiest authentication scheme to implement, but also the most insecure. You should take precautions in the design of your application to ensure your key is not compromised.

```
https://api.domaintools.com/v1/domaintools.com?api_username=example&api_key=xxxxxx
```

Required Parameters	Value
api_username	Your API username
api_key	Your API key

2.4 API Key (Header) Authentication

Authenticate your requests by including the API key in the header of each HTTP request. The API key serves as a unique identifier and is used to authenticate your requests.

2.4.1 Required API Key Authentication Parameters

Required Parameters	Value
X-Api-Key	MY_API_KEY

2.4.2 Example Request with API Key Authentication

```
curl -H 'X-API-Key: MY_API_KEY'
'https://api.domaintools.com/v1/feed/nod/?after=-60'
```

2.5 HMAC Signed Authentication

HMAC, or hashed message authentication code, is our preferred authentication scheme. It follows the principles outlined in RFC2104 and provides a straightforward but secure method of protecting your API key.

It involves passing a hash composed of your api username, the current date and time, and the request URI. This hash is then signed with your authentication key. The result is a request that expires after a brief period of time and, most importantly, does not contain your authentication key.

MD5, SHA-1, and SHA-256 are supported HMAC hashing algorithms. We recommend SHA256 HMAC, as a successor of SHA1. Although NIST policy on hash functions still allows HMAC SHA-1, see [Hash Functions in the NIST CSRC](#) page. Overall, SHA-256 HMAC is also a more secure (ex, against brute force attacks) alternative to MD5 HMAC.

Please check your own compliance obligations, or guidance, as to what cryptographic functions should be used. For example, see the [NIST FIPS 140-2 Guidance](#) for more information for organizations that must use FIPS 140-2 compliant algorithms.

2.5.1 HMAC-Signed Authentication URI/Parameters

Required Parameters	Value
api_username	Your API username
timestamp	Current timestamp, in ISO 8601 format. Timestamps should contain the timezone offset, like below: 2020-02-01T14:37:59-0800 2020-02-01T22:37:59Z 2020-02-02T10:37:59+1200
signature	HMAC signature of your request. SHA-256 HMAC is recommended. MD5 HMAC and SHA-1 HMAC are supported. Only sign the URL path.

2.5.2 Creating a HMAC-signed authentication request

To create the signature, build a string with your API username, the timestamp in ISO 8601 format, and the URI you are requesting. That string is then signed with your API key using an HMAC function to generate the signature. The steps are below.

1. Add your api_username.
The API username is the account that has been provisioned access to the relevant DomainTools API product (aka API endpoint). If this username does not have access rights, an error message will be returned.
2. Associate a timestamp.
The timestamp value for the hash and timestamp parameter need to be identical. Make sure your server time is accurate and always use a fresh timestamp.

3. Add the request URI.
Please see the individual API product User Guides for the endpoint's URI format.
4. Add the host.
The host is the API endpoint server. In this case, it is api.domaintools.com.
5. Use the appropriate HMAC algorithm.
MD5, SHA-1, or SHA-256 for the hashing algorithm are supported. See the HMAC Algorithms section for more information.

The result is a request that expires after a brief period of time and, most importantly, does not contain your authentication key in the clear.

The resulting URLs will generally be similar, regardless of whichever programming language is used.

2.5.3 Security Note

Ensure that your current language version supports the required HMAC (and other relevant) extensions and libraries. For example, a failure in HMAC-signed URL will result in a URL with no signature value, such as below:

```
http://api.domaintools.com/v1/yourdomain.com/whois
```

3 Iris Rate Limits

This document explains rate limits and quota consumption for Iris Investigate, Iris Enrich, and Iris Detect in both the API and UI.

3.1 Check your rate limits

Use the following methods to determine the current rate limit associated with Iris Investigate, Iris Enrich, and Iris Detect API endpoints:

1. /account Endpoint

- Querying the Account Information endpoint provides details on query limits, usage, and expiration dates for all licensed Iris endpoints.
- For more information, refer to the [Account Information endpoint documentation](#).

2. API Admin Dashboard

- The API Admin can access their API dashboard to view query limits, usage, and expiration dates for all licensed endpoints.
- To access the API dashboard, visit the [DomainTools research page](#), select the account dropdown menu, and click on API Admin.

3.2 Free test queries

Multiple endpoints, including Iris Detect and Investigate, offer free test queries that don't count against your quota. Use these domains for testing and development:

- domaintools.com
- dnsdb.info
- example.com

3.2.1 Endpoints with free test queries

The following endpoints support free test queries with the domains listed above:

[Domain Profile](#), [Domain Reputation](#), [Hosting History](#), [Iris Enrich and Investigate](#), [Reverse IP](#), [WHOIS history](#), [WHOIS Lookups](#)

Using these test domains allows you to verify API integration and test functionality without consuming your quota.

3.3 Iris Investigate rate limits and quota consumption

The system measures quotas at the group level and resets them each month.

3.3.1 Duplicate query policies

Iris Investigate treats identical queries differently depending on whether you use the UI or API:

3.3.1.1 User Interface

Identical queries within 30 days don't count against your quota. This includes:

- All searches (omnisearch and advanced search)
- Search hash queries
- Search hash reloading
- Search node revisits

A query is considered identical if it has the same filters, sorting, and pagination parameters.

3.3.1.2 API

Regular queries: Identical queries within 1 hour don't count against quota. A query is considered identical if it has the same filters, sorting, and pagination parameters as a previous query made within the last hour.

Search hash queries: Identical search hash queries within 10 minutes don't count against quota. A search hash query uses the `search_hash` parameter to retrieve previously saved search results.

3.3.2 Activities that consume quota

3.3.2.1 Pivot Engine Queries in the Iris Investigate UI

The following activities consume your quota:

- Executing an omnisearch (from landing or search pages) that returns results
- Executing an advanced search that returns results
- Sending a result to the Pivot Engine (including narrow, expand, new, and exclude functions)
- Revisiting a search node more than 30 days since it was created
- Loading new pages in the Pivot Engine
- Sorting Pivot Engine results

For information on duplicate queries that don't consume quota, see [Duplicate query policies](#).

3.3.2.2 Passive DNS (pDNS) Queries in the Iris Investigate UI

The following activities consume your quota:

- Executing a search query that returns results (from either the search field or popovers throughout the UI)
- Executing/triggering the load more (infinite scroll) function in search results
- Revisiting a search node more than 30 days since it was created

For information on duplicate queries that don't consume quota, see [Duplicate query policies](#).

3.3.2.3 Queries in the Iris Investigate API

The following activities consume your quota:

- Executing a query
- Loading additional pages of results

For information on duplicate queries that don't consume quota, see [Duplicate query policies](#).

3.4 Iris Enrich rate limits

Iris Enrich uses an independent service level with its own rate limits, separate from Iris Investigate. This independence allows Iris Enrich to be used at much greater scale and throughput than Iris Investigate.

Key differences:

- **Independent quota:** Enrich queries don't count toward your Iris Investigate quota
- **Separate service level:** Rate limits are defined independently for Enrich
- **Higher throughput:** Optimized for high-volume domain enrichment use cases
- **No pivot parameters:** Unlike Investigate, Enrich is optimized for straightforward domain enrichment rather than pivoting

Your Enrich service level, query caps, and rate limits are configured independently on your enterprise account. For complete details on using Iris Enrich, consult the [Iris Enrich API documentation](#).

3.5 Iris Detect rate limits

Iris Detect has specific endpoint rate limits designed for monitoring and triage workflows.

3.5.1 Domain endpoints rate limits

The domains endpoints (/domains/new and /domains/watched) have a rate limit of **1 query per hour** for each endpoint.

Pagination: Requests using the offset parameter to retrieve additional pages of results aren't restricted by the hourly limit. If your hourly query returns more domains than the response limit, you can retrieve the complete result set using pagination without waiting.

3.5.2 Development and testing

During integration and testing, use the preview parameter to bypass rate limits:

- Including preview=1 in requests limits responses to 2 domains
- Allows up to 30 requests per minute
- Enables rapid iteration during development

3.5.3 Other endpoints

Monitor management endpoints and domain triage actions (watchlist, ignore, escalate) have standard API rate limits. For complete details on Iris Detect API endpoints and usage, consult the [Iris Detect API documentation](#).

4 Iris API Error Codes

The Iris API uses standard HTTP status codes to indicate the success or failure of requests. Understanding these codes helps you handle errors gracefully and troubleshoot issues effectively.

4.1 Status Codes

Status Code	Description
200	OK - The request was successful.
206	Partial Content - The request was partially successful. Some data is unavailable but the request returned what was available. This typically occurs when backend services are temporarily unavailable or when data for some requested items cannot be retrieved.
400	Bad Request - The request is malformed or contains invalid parameters. Check the error message for details about what needs to be corrected.
401	Unauthorized - Authentication credentials are missing, invalid, or expired. Verify your API key and authentication method.
403	Forbidden - The authenticated account does not have permission to access this resource or endpoint. This may indicate insufficient subscription level or access rights.
404	Not Found - The requested resource does not exist. This may occur when querying for a domain that has never been registered or for data that is not available in our systems.
500	Internal Server Error - An unexpected error occurred on the server. If this persists, contact DomainTools support at enterprisesupport@domaintools.com .
503	Service Unavailable - The service is temporarily unavailable, typically due to maintenance or high load. Implement exponential backoff and retry the request.

4.2 Error Response Format

Error responses follow a consistent JSON structure:

```
{
  "error": {
    "code": 400,
    "message": "Invalid domain name format"
  }
}
```

```
}  
}
```

4.3 Common Error Scenarios

4.3.1 Authentication Errors (401)

Cause: Missing or invalid API credentials.

Solutions:

- Verify your API key is correct
- Check that your authentication method (header, HMAC, or basic auth) is properly configured
- Ensure your API key hasn't expired

4.3.2 Authorization Errors (403)

Cause: Insufficient permissions or subscription level.

Solutions:

- Verify your account has access to the requested endpoint
- Check your subscription includes the API product you're trying to use
- Contact your account manager if you need additional access

4.3.3 Not Found Errors (404)

Cause: Requested resource doesn't exist.

Common scenarios:

- Querying a domain that has never been registered
- Requesting historical data that predates our records
- Using an invalid endpoint path

4.3.4 Rate Limiting (503)

Cause: Too many requests in a short time period.

Solutions:

- Implement exponential backoff retry logic
- Check your rate limits using the Account Information endpoint
- Distribute requests over time rather than in bursts
- Consider upgrading your subscription for higher rate limits

4.4 Best Practices

1. **Always check status codes:** Don't assume success - verify the response status code
2. **Parse error messages:** Error messages contain specific details about what went wrong
3. **Implement retry logic:** Use exponential backoff for 500 and 503 errors
4. **Log errors:** Keep detailed logs of errors for troubleshooting
5. **Monitor rate limits:** Track your API usage to avoid hitting limits

4.5 Iris-Specific Notes

4.5.1 Partial Content (206)

The Iris Enrich and Investigate endpoints may return a 206 status code when:

- Some backend data sources are temporarily unavailable
- Data for certain requested domains cannot be retrieved
- The response contains partial results rather than complete data

When you receive a 206 response, the returned data is still valid and usable - it simply indicates that some information is missing.

4.5.2 Detect Endpoints

Iris Detect endpoints (watchlists and monitors) use a subset of the standard error codes:

- 200 (OK)
- 400 (Bad Request)
- 401 (Unauthorized)
- 403 (Forbidden)
- 404 (Not Found)
- 422 (Unprocessable Entity) — write endpoints only

These endpoints do not return 500 or 503 errors in normal operation.

4.5.3 Validation Errors (422)

Iris Detect write endpoints (POST, PUT, and PATCH) return a 422 Unprocessable Entity status code when the request body is syntactically valid JSON but contains values that fail server-side validation. Unlike the standard error format, the 422 response includes per-field validation messages:

```
{
  "error": {
    "code": 422,
    "summary": "The given data was invalid.",
    "messages": {
      "term": ["The term field is required."],
      "watchlist_domain_ids": ["The watchlist domain ids field is required."]
    }
  },
  "resources": {
    "support": "https://docs.domaintools.com"
  }
}
```

Common validation triggers include:

- Missing required fields (for example, term when creating a monitor)
- Values that don't meet minimum length requirements (for example, term must be at least 3 characters)
- Invalid enum values (for example, an unrecognized state value)

This applies to the following endpoints:

Endpoint	Method
/v1/iris-detect/domains/	PATCH
/v1/iris-detect/escalations/	POST
/v1/iris-detect/monitors/	POST
/v1/iris-detect/monitors/	PUT

Iris Enrich and Iris Investigate endpoints do not return 422. These endpoints return 400 for invalid parameters.

5 Iris Detect API guide

5.1 Overview

The Iris Detect API enables automated workflows for discovering and triaging lookalike domains that impersonate your brands, partners, or infrastructure. Retrieve newly discovered domains, query watched domains, and programmatically escalate threats for blocking or submission to Google Web Risk. For a complete overview of Iris Detect capabilities and use cases, see the [Iris Detect overview](#).

Iris Detect protects against malicious domains impersonating your brands and domains, safeguarding your organization, customers, and trademarks. You can also defend against supply chain attacks where malicious domains impersonate well-known technology vendors or partners you work with regularly.

Iris Detect discovers new domains appearing globally that mimic your brands. You can quickly see key information from DNS, WHOIS/RDAP, and the DomainTools Risk Score to assess the level of threat. Iris Detect also monitors domains over time so you can see how they evolve and take action as needed.

5.2 About Iris Detect APIs

The Detect API endpoints automate workflows and feed data from Iris Detect into other applications. Many actions performed via the Detect UI can be programmatically scheduled for convenience.

5.2.1 Functionality supported via APIs

- **Read:** Retrieve data about discovered domains, monitors, and escalations
- **Write:** Make changes such as adding a domain to the watchlist or escalating a domain for further action
- **Monitor management:** Create, update, and delete monitors programmatically

5.2.2 Not supported by API

- **Event-based updates:** There is no way to “subscribe” to changes. All updates must be retrieved by “pull” action. However, you can schedule email alerts for regular updates within the UI.

5.3 Getting started

Iris Detect requires an Enterprise account login on DomainTools, and your account must include Iris Detect access. Additionally, user permissions control whether users can manage monitors, triage domains, or escalate domains. By default, users have read-only access to the application.

To use the Detect APIs, you need your username and an API key.

5.4 Authentication

Consult the [Iris API authentication options](#) for details on the available authentication methods.

5.5 Authorization and permissions

API accounts require specific permissions to perform domain triage actions. These permissions must be configured by your account administrator.

5.5.1 Permission types

Depending on your account configuration, you may have different levels of access:

- **Read-only permissions:** The account can only retrieve discovered domains but cannot perform triage actions. Accounts with read-only permissions don't have access to the add/remove to watchlist endpoint or escalation endpoints.
- **Read + write permissions:** The account has full access to perform domain triage actions including:
 - Adding domains to the watchlist
 - Ignoring domains
 - Escalating domains (for internal blocking or to Google Safe Browsing)

5.5.2 Required permissions for API actions

To perform domain triage actions via the API, your API account must have the following permissions configured:

- **Add to watchlist:** Required to add domains to your watchlist for ongoing monitoring
- **Ignore:** Required to mark domains as false positives and remove them from active monitoring
- **Escalate:** Required to escalate domains for internal blocking or to Google Safe Browsing
- **Manage monitors:** Required to create, update, or delete monitors via the API

If you need these permissions enabled for your API account, contact your account administrator.

5.6 API rate limits

5.6.1 Endpoints with special rate limits

- **Domains:** The Domains endpoints can be queried with a frequency of up to once per hour for each the `domains/new` and `domains/watched` calls.
- **Pagination requests:** Requests to get full domain lists are not restricted. For example, if the hourly call for new domains returns more than the `limit` of domains returned in a single response, the additional domains can be queried using the `offset` variable to complete the results without being impacted by the hourly restriction.

5.6.2 Bypassing rate limits during integration work

During integration of the domain API endpoints, including the `preview` parameter in requests limits the responses to just 2 domains but allows for up to 30 requests per minute. This enables rapid iteration during development and testing. (Use the `preview` parameter for development and testing only. Don't use it for sustained production queries. Sustained usage violates our Terms of Service and can result in service suspension.)

For complete information about Iris API rate limits across all products, consult the [Iris API rate limits](#) documentation.

DomainTools My Account Image

Figure 1: DomainTools My Account Image

5.7 Finding and resetting the API key

To find or reset the DomainTools API key via <https://research.domaintools.com/>:

Login to the DomainTools API Dashboard. Only the API owner account can reset the API key. There are two places to access this:

- Select the **Account** menu and select the **API Admin** menu item to access the API Dashboard.
 - In the **My Account** section, select the **View API Dashboard** link that is located in the Account Summary tab.
1. In the **API Product Information** tab, go to the section called API Username to obtain your username.
 2. In the same tab, you can find your **API Key**.
 3. To reset your API key:
 - Select the **Reset API Key**.
 - Creating a new key will cause the existing key to be discarded immediately. This action cannot be undone.
 - Any requests that are made with the old key will no longer validate for this user.

5.8 API endpoints

5.8.1 Monitor list

This endpoint allows users to retrieve the monitors and their respective IDs for your account. The monitor IDs can then be used to retrieve newly discovered or recently changed domains for individual monitors using the `monitor_id` variable.

You can also request new or changed domains across all monitors in a single request by not specifying the `monitor_id` variable. Since the endpoint is limited to 1 query request per hour, making a request for all monitors retrieves results for all monitors every hour. The monitor endpoint can still be used to help map domains to their monitor if that is important for the integration.

The `include_counts` variable shows the number of domains associated with the various categories such as New, Changed, etc. for each monitor.

You can sort and order the results with options such as term, creation date, created by, and others.

5.8.1.1 Endpoint URL

`https://api.domaintools.com/v1/iris-detect/monitors`

5.8.1.2 Method

GET

5.8.1.3 Conditionally required parameters

Name	Type	Valid values	Default values	Notes
datetime_counts_since	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Conditionally required if the <i>include_counts</i> parameter is included.

5.8.1.4 Optional parameters

Name	Type	Valid values	Default values	Notes
include_counts	bool		false	Includes counts for each monitor for new, watched, changed and escalated domains
sort	string[]	term, created_date, domain_counts_changed, domain_counts_discovered	term	Provides options for sorting the monitor list.
order	string	asc, desc	desc	Toggles ordering for selected sort option.
offset	int	0-100000	0	For paginating requests beyond the limit.
limit	int	500		Restricted to maximum 100 if include_counts=true

5.8.1.5 Example usage

5.8.1.5.1 Retrieve all monitors with counts

GET https://api.domaintools.com/v1/iris-detect/monitors/?datetime_counts_since=2022-01-14%2016:27:31&include_counts=true &api_username=<username>&api_key=<API key>

5.8.2 Create monitor

Create a new monitor to track lookalike domains for a specific term. The monitor begins discovering matching domains immediately after creation.

Requires the “manage monitors” permission.

5.8.2.1 Endpoint URL

<https://api.domaintools.com/v1/iris-detect/monitors/>

5.8.2.2 Method

POST

5.8.2.3 Required parameters (JSON body)

Name	Type	Description
term	string	The keyword to monitor for domain variations.

5.8.2.4 Optional parameters (JSON body)

Name	Type	Default	Description
match_substring_variations	boolean	false	When true, includes domains where variations of the term appear as substrings within matched domains. See how domain matching works .
nameserver_exclusions	string[]	[]	Nameserver patterns to exclude. Wildcards (*) are accepted (for example, *.domaintools.com). A domain is excluded only when all of its nameservers match an exclusion pattern.
text_exclusions	string[]	[]	Text strings to exclude from results. Domains containing any of these strings are excluded.

5.8.2.5 Example usage

5.8.2.5.1 Create a monitor with exclusions

POST https://api.domaintools.com/v1/iris-detect/monitors/?api_username=<username>&api_key=<API key>

Body:

```
{
  "term": "domaintools",
  "match_substring_variations": true,
  "text_exclusions": ["selection", "redomaintools"],
  "nameserver_exclusions": ["*.domaintools.com"]
}
```

```
}
```

5.8.2.6 Response

The response includes a status field and the created monitor object:

```
{
  "status": "created",
  "monitor": {
    "term": "domaintools",
    "match_substring_variations": true,
    "nameserver_exclusions": ["*.domaintools.com"],
    "text_exclusions": ["selection", "redomaintools"],
    "id": "oab3nQ7m8B",
    "created_date": "2026-03-02T18:33:04.245472Z",
    "updated_date": "2026-03-02T18:33:04.245472Z",
    "state": "active",
    "status": "pending",
    "created_by": "<username>"
  }
}
```

The monitor status is pending immediately after creation and changes to completed once initial domain discovery finishes.

5.8.3 Update monitor

Update the configuration of an existing monitor. You can modify match settings and exclusions without recreating the monitor.

Requires the “manage monitors” permission.

Note

A recently created monitor may not be immediately available for updates. Wait until the monitor’s status changes from pending to completed before sending update requests.

5.8.3.1 Endpoint URL

<https://api.domaintools.com/v1/iris-detect/monitors/>

5.8.3.2 Method

PUT

5.8.3.3 Required query parameters

Name	Type	Description
monitor_id	string	The ID of the monitor to update. Retrieve monitor IDs from the monitor list endpoint.

5.8.3.4 Optional parameters (JSON body)

Name	Type	Description
match_substring_variations	boolean	Enable or disable substring variation matching.
nameserver_exclusions	string[]	Replace the current nameserver exclusion list. Wildcards (*) are accepted.
text_exclusions	string[]	Replace the current text exclusion list.

Include only the fields you want to change. Omitted fields retain their current values.

5.8.3.5 Example usage

5.8.3.5.1 Enable substring matching on an existing monitor

PUT https://api.domaintools.com/v1/iris-detect/monitors/?monitor_id=JKwjv5Xp8x&api_username=<username>

Body:

```
{
  "match_substring_variations": true
}
```

5.8.3.5.2 Update text exclusions

PUT https://api.domaintools.com/v1/iris-detect/monitors/?monitor_id=JKwjv5Xp8x&api_username=<username>

Body:

```
{
  "text_exclusions": ["selection", "pineapple"]
}
```

5.8.4 Delete monitor

Permanently delete a monitor and stop tracking its associated domains.

Requires the “manage monitors” permission.

5.8.4.1 Endpoint URL

<https://api.domaintools.com/v1/iris-detect/monitors/>

5.8.4.2 Method

DELETE

5.8.4.3 Required query parameters

Name	Type	Description
monitor_id	string	The ID of the monitor to delete. Retrieve monitor IDs from the monitor list endpoint.

5.8.4.4 Request body

None.

5.8.4.5 Example usage

5.8.4.5.1 Delete a monitor

DELETE https://api.domaintools.com/v1/iris-detect/monitors/?monitor_id=JKwjev5Xp8x&api_username=<us

5.8.4.6 Response

The response confirms the deletion and returns the monitor ID:

```
{
  "status": "deleted",
  "monitor": {
    "id": "JKwjev5Xp8x"
  }
}
```

5.8.5 Domains

The Domains endpoint enables users to retrieve details associated with domains for a specific monitor or all monitors. The endpoint can be used to view:

- Newly discovered domains
- Recently changed domains (with changes to key DNS and RDAP/WHOIS data tracked by the application)
- Domains that have been recently escalated in the application, with the escalations being for internal blocking or for sending the domains to Google's Web Risk service.
- Escalations can be triggered either directly in the application UX or via API (covered in the Escalation section below).

A number of optional parameters are made available to filter the results as needed.

The `/new` endpoint can provide newly discovered domains for monitors created in Iris Detect.

The `/watched` endpoint provides recently changed domains for the domains from a monitor's list of watched domains. (Domains can be selected for watching from either the Iris Detect UX or via API, discussed below.)

The `/watched` endpoint can also be used to query for domains that have been escalated for blocking or to Google Safe browsing. (Domains can be escalated either from the Iris Detect UX or via API, discussed below.)

5.8.5.1 Endpoint URL

<https://api.domaintools.com/v1/iris-detect/domains/new/>

<https://api.domaintools.com/v1/iris-detect/domains/watched/>

5.8.5.2 Method

GET

5.8.5.3 Required parameters

None

5.8.5.4 Optional parameters

Name	Type	Valid values	Default values	Notes
monitor_id	string			Monitor ID from the monitors response – only used when requesting domains for specific monitors.
escalation_types	string[]	blocked, google_safe		Filters on specific escalation types
tlds	string[]	Use tlds[]= for each TLD. Do not combine TLDs in a single tlds[] parameter.		Filters on one or multiple TLDs
risk_score_ranges	string[]	0-0, 1-39, 40-69, 70-99, 100-100		Filters on domains with a risk score in different ranges.
mx_exists	boolean			Whether domain currently has an MX record in DNS.
domain_state	string	active, inactive		
discovered_before	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Can be used with <code>discovered_since</code> to define a complete date/time window. Exclusive.

Name	Type	Valid values	Default values	Notes
discovered_since	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Can be used with <code>discovered_before</code> . Relevant for the <code>/new</code> endpoint to control the timeframe for when a new domain was discovered. Inclusive.
changed_before	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Most relevant for the <code>/watched</code> endpoint to control the timeframe for changes to DNS or WHOIS/RDAP fields for watched domains.
changed_since	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Most relevant for the <code>/watched</code> endpoint to control the timeframe for changes to DNS or WHOIS/RDAP fields for watched domains.
escalated_since	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Most relevant for the <code>/watched</code> endpoint to control the timeframe for when a domain was most recently escalated.
search	string			A “contains” search for domain name.
sort	string[]	discovered_date, changed_date, risk_score		Sort order of domain list response.
order	string	asc, desc		Ordering options for sorting
include_domain_data	boolean			Includes DNS and RDAP/WHOIS details.

Name	Type	Valid values	Default values	Notes
screenshots	flag	1		Set to 1 to include screenshot data in the response. See working with screenshots for details.
offset	int	0-100000	0	For paginating requests beyond the response limit. <i>No other parameter values can change when paginating.</i>
limit	int	0-100 or 0-50 if include-domain-data=true	0	Limits result size.
preview	boolean			Use during API implementation and testing. Including with value = 1 will limit results to 2 but not be limited by hourly restrictions. For development and testing only; sustained usage violates Terms of Service.

5.8.5.5 Example usage

Note that date/time format follows ISO-8601. Not including a timezone indicator will be assumed to be in Pacific Time Zone.

5.8.5.5.1 Retrieve newly discovered domains since a specific date/time for Pacific Standard Time with millisecond precision

GET https://api.domaintools.com/v1/iris-detect/domains/new/?discovered_since=2022-01-12%2000:00.000000%20-08:00&api_username=<username>&api_key=<API key>

5.8.5.5.2 Retrieve all watched domains that changed within a given set of dates

GET https://api.domaintools.com/v1/iris-detect/domains/watched/?api_key=<api_key>&api_username=<username>&discovered_before=2024-01-01

5.8.5.5.3 Retrieve domain that have been escalated for internal blocking since a specific date/time for Pacific Standard Time

GET [https://api.domaintools.com/v1/iris-detect/domains/watched/?escalated_since=2022-01-11%2016:00:00%20-08:00&api_username=<username>&api_key=<API key>&escalation_types\[\]=blocked](https://api.domaintools.com/v1/iris-detect/domains/watched/?escalated_since=2022-01-11%2016:00:00%20-08:00&api_username=<username>&api_key=<API key>&escalation_types[]=blocked)

5.8.5.5.4 Retrieve domains when results exceed response limit of 100 results using offset parameter

- Example shows subsequent call for second group of 100 domains

GET https://api.domaintools.com/v1/iris-detect/domains/new/?offset=100&api_username=<username>&api_key=<API key>

5.8.6 Ignored domains

The Ignored Domains endpoint enables users to retrieve details about domains that have been marked as ignored (false positives) across their monitors. When a domain is marked as ignored, it's removed from active monitoring and no longer generates alerts. This endpoint allows you to review ignored domains and, if needed, move them back to your watchlist.

Like the other domain endpoints, you can retrieve ignored domains for a specific monitor or across all monitors. Various optional parameters are available to filter and search the results.

5.8.6.1 Endpoint URL

<https://api.domaintools.com/v1/iris-detect/domains/ignored/>

5.8.6.2 Method

GET

5.8.6.3 Required parameters

None

5.8.6.4 Optional parameters

Name	Type	Valid values	Default values	Notes
monitor_id	string			Monitor ID from the monitors response – only used when requesting domains for specific monitors.
tlds	string[]	Use tlds[]= for each TLD. Do not combine TLDs in a single tlds[] parameter.		Filters on one or multiple TLDs
risk_score_ranges	string[]	0-0, 1-39, 40-69, 70-99, 100-100		Filters on domains with a risk score in different ranges.

Name	Type	Valid values	Default values	Notes
mx_exists	boolean			Whether domain currently has an MX record in DNS.
domain_state	string	active, inactive		
discovered_before	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Can be used with <code>discovered_since</code> to define a complete date/time window. Exclusive.
discovered_since	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Can be used with <code>discovered_before</code> to control the timeframe for when a domain was discovered. Inclusive.
changed_since	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Filters domains based on when they were last changed.
escalated_since	datetime OR integer	ISO-8601 date/time format for absolute time OR a negative integer for the (relative) number of seconds ago.		Filters domains based on when they were most recently escalated.
search	string			A “contains” search for domain name.
sort	string[]	discovered_date, changed_date, risk_score		Sort order of domain list response.
order	string	asc, desc		Ordering options for sorting
include_domain_data	boolean			Includes DNS and RDAP/WHOIS details.

Name	Type	Valid values	Default values	Notes
offset	int	0-100000	0	For paginating requests beyond the response limit. <i>No other parameter values can change when paginating.</i>
limit	int	0-100 or 0-50 if include-domain-data=true	0	Limits result size.
preview	boolean			Use during API implementation and testing. Including with value = 1 will limit results to 2 but not be limited by hourly restrictions. For development and testing only; sustained usage violates Terms of Service.

5.8.6.5 Example usage

5.8.6.5.1 Retrieve all ignored domains

GET https://api.domaintools.com/v1/iris-detect/domains/ignored/?api_username=<username>&api_key=<API key>

5.8.6.5.2 Retrieve ignored domains for a specific monitor

GET https://api.domaintools.com/v1/iris-detect/domains/ignored/?monitor_id=JKwjv5Xp8x&api_username=<username>

5.8.6.5.3 Retrieve ignored domains discovered within a specific timeframe

GET https://api.domaintools.com/v1/iris-detect/domains/ignored/?discovered_since=2023-01-01&discovered_before=2024-01-01&api_username=<username>&api_key=<API key>

5.8.6.5.4 Search for specific ignored domains

GET https://api.domaintools.com/v1/iris-detect/domains/ignored/?search=example&api_username=<username>

5.8.7 Add and remove from watchlist

Analyzing and triaging newly discovered domains is an activity that should be done regularly, so New domains only show recently discovered domains. Triage options include:

- **Add to watchlist:** Adding a domain to the Watchlist removes it from New and allows tracking of when domains most recently changed in DNS or WHOIS/RDAP. Watched domains will have screenshots captured daily so you see how the webpage evolves over time.

- **Ignore:** If a domain is obviously a false positive, Ignoring the domain removes it from New and places it under Ignored. The decision can always be reversed and the domain can be moved from Ignored to Watched instead.

5.8.7.1 Endpoint URL

<https://api.domaintools.com/v1/iris-detect/domains/>

5.8.7.2 Method

PATCH

5.8.7.3 Required parameters

Name	Type	Valid values	Default value	Notes
watchlist_domain_ids	string[]			ID(s) of domains being triaged.
state	string	watched, ignored		Add domains to Watchlist or ignore and mute alerts for those domains.

5.8.7.4 Optional parameters

None

5.8.7.5 Example usage

5.8.7.5.1 Add domain to watchlist URL:

https://api.domaintools.com/v1/iris-detect/domains/?api_username=<username>&api_key=<API key>

Body:

```
{
  "watchlist_domain_ids": ["<domain ID>"],
  "state": "watched"
}
```

5.8.7.5.2 Remove domain from watchlist URL

https://api.domaintools.com/v1/iris-detect/domains/?api_username=<username>&api_key=<API key>

Body:

```
{
  "watchlist_domain_ids": ["<domain ID>"],
  "state": "ignored"
}
```

5.8.8 Escalate

Two escalation activities are supported for the API as shown below.

Escalating a New domain will also add it to be Watched for changes.

- Domains can be sent to Google Web Risk. If Google agrees the domain is malicious, it will be blocked in Chrome browsers globally. This list is also picked up by Apple for their Safari browser and Firefox.
- Domains can be marked for Blocking if they are to be blocked in internal network defense infrastructure. Loading a domain into a company's firewalls and other systems needs to take place via API query combined with a system action on the customer end.
- Limits for escalation request API calls
 - Blocked domains: A maximum of 100 domains per request.
 - Google Web Risk: A maximum of 10 domains per request.
 - There is a limit of 30 requests per minute (each request can include multiple domains per above).

5.8.8.1 Endpoint URL

<https://api.domaintools.com/v1/iris-detect/escalations/>

5.8.8.2 Method

POST

5.8.8.3 Required parameters

Name	Type	Valid values	Default value	Notes
watchlist_domain_ids	string[]			ID(s) of domains to be escalated
escalation_type	string	blocked, google_safe		

5.8.8.4 Optional parameters

None

5.8.8.5 Example usage

5.8.8.5.1 Escalate to Google Safe Browsing URL:

https://api.domaintools.com/v1/iris-detect/escalations/?api_username=<username>&api_key=<API key>

Body:

```
{
  "watchlist_domain_ids": ["<domain ID>"],
  "escalation_type": "google_safe"
}
```

5.8.8.5.2 Escalate for internal blocking URL:

https://api.domaintools.com/v1/iris-detect/escalations/?api_username=<username>&api_key=<API key>

Body:

```
{
  "watchlist_domain_ids": ["<domain ID>"],
  "escalation_type": "blocked"
}
```

5.9 Working with screenshots

You can retrieve domain screenshots through the Iris Detect API by adding the `screenshots=1` parameter to your domain requests. For detailed information about retrieving, resizing, and interpreting screenshot data, see [working with screenshots in Iris APIs](#).

5.10 Appendix: API response examples

5.10.1 Monitor list example

This is a sample of the response from a query to the Monitor List endpoint, preceded by the request:

5.10.1.1 Request example

```
GET https://api.domaintools.com/v1/iris-detect/monitors/?datetime_counts_since=2022-01-20%2011:03:21&include_counts=true&api_username=<username>&api_key=<API key>
```

5.10.1.2 Response example

This is a sample of the response, including counts for each monitor:

```
{
  "total_count": 16,
  "offset": 0,
  "limit": 500,
  "monitors": [
    {
      "term": "domaintools",
      "match_substring_variations": true,
      "nameserver_exclusions": [],
      "text_exclusions": [],
      "id": "JKwjv5Xp8x",
      "created_date": "2022-01-15T00:32:04.418134+00:00",
      "updated_date": "2022-01-15T00:32:04.418134+00:00",
      "state": "active",
      "status": "completed",
      "domain_counts": {
        "new": 2,
        "watched": 5,
        "escalated": 1,
        "changed": 2
      },
      "created_by": "<username>"
    }
  ]
}
```

5.10.2 Domain list example

The following example is for querying domains/new endpoint, the results would be the same for the domains/changed endpoint for querying either changed domains or domains that have been escalated since a given date/time.

5.10.2.1 Request example

The request includes the boolean variable to include domain data for key fields for DNS and WHOIS/RDAP.

GET https://api.domaintools.com/v1/iris-detect/domains/new/?discovered_since=2022-01-20%20%2009:40:12.000000%20-08:00&api_username=<username>&api_key=<API key>&include_domain_data=

5.10.2.2 Response example

This is an example of the response, including details for DNS and WHOIS/RDAP data. Note the "risk_score_status": "provisional" field used to indicate a partially example of the response, including details for DNS and WHOIS/RDAP data. The risk score status indicates whether the scoring is provisional or full. Newly discovered domains will have initial proximity or phishing scores within a few minutes and the score is designated as provisional. Full risk scoring (across all 4 algorithms including malware and spam) is typically available within 15-20 minutes of discovery. Risk scores are also updated when significant changes are detected to a domain's DNS records or other attributes. All active domains continue to be scored daily.

```
{
  "watchlist_domains": [
    {
      "state": "new",
      "domain": "bestanbanks.in",
      "status": "active",
      "discovered_date": "2022-01-20T20:26:14.378000+00:00",
      "changed_date": "2022-01-20T20:27:58.000000+00:00",
      "risk_score": 67,
      "risk_score_status": "provisional",
      "risk_score_components": {
        "proximity": 21,
        "threat_profile": {
          "phishing": 67
        }
      }
    },
    "mx_exists": true,
    "tld": "in",
    "id": "8avjyB1Xaj",
    "escalations": [
      {
        "escalation_type": "blocked",
        "id": "p6db6jQbVJ",
        "created": "2022-01-07T16:44:40.491903+00:00",
        "created_by": "<username>"
      }
    ],
    "monitor_ids": [
      "Y41pGXBjEq"
    ],
    "name_server": [
      {
        "host": "dimitris.ns.cloudflare.com"
      },
      {
        "host": "zainab.ns.cloudflare.com"
      }
    ],
    "registrant_contact_email": [
      "please contact the registrar listed above"
    ],
    "registrar": "GoDaddy.com, LLC",
    "create_date": 20220120,
    "ip": [
      {
```

```

        "country_code": "US",
        "ip": "172.67.192.30",
        "isp": "CloudFlare Inc."
    },
    {
        "country_code": "US",
        "ip": "104.21.81.241",
        "isp": "CloudFlare Inc."
    }
],
"mx": [
    {
        "host": "mx1.hostinger.in"
    },
    {
        "host": "mx2.hostinger.in"
    }
]
}
},
"total_count": 1,
"count": 1,
"offset": 0,
"limit": 100
}

```

5.10.3 Add domain to be watched example

This request is for adding a domain to the account's Watchlist. A similar request is used to ignore domains, using the "state" : "ignored" instead of "state" : "watched".

5.10.3.1 Request example

Request URL:

PATCH https://api.domaintools.com/v1/iris-detect/domains/?api_username=<username>&api_key=<API key>

Request Body:

```

{
  "watchlist_domain_ids": ["AaLMA1o0E0"],
  "state": "watched"
}

```

5.10.3.2 Response example

```

{
  "watchlist_domains": [
    {
      "state": "watched",
      "domain": "facebook-09835.pl",
      "discovered_date": "2022-01-20T18:35:29.801000+00:00",
      "changed_date": "2022-01-20T18:57:34.000000+00:00",
      "id": "AaLMA1o0E0",
      "assigned_by": "<username>",
      "assigned_date": "2022-01-20T22:27:39.000000+00:00"
    }
  ]
}

```

```
    ]  
  }  
}
```

5.10.4 Escalate a domain for blocking example

This request is for escalate a domain for internal blocking. A similar request is used to escalate domains to Google Safe Browsing using the "escalation_type" : "google_safe" instead of "escalation_type" : "blocked".

5.10.4.1 Request example

URL requested:

POST https://api.domaintools.com/v1/iris-detect/escalations/?api_username=<username>&api_key=<API

Request body:

```
{  
  "watchlist_domain_ids": ["AaLMA1g0E0"],  
  "escalation_type": "blocked"  
}
```

5.10.4.2 Response example

```
{  
  "escalations": [  
    {  
      "watchlist_domain_id": "AaLMA1g0E0",  
      "escalation_type": "blocked",  
      "id": "BwdgkK8g6p",  
      "created_date": "2022-01-20T23:25:48.990908+00:00",  
      "updated_date": "2022-01-20T23:25:48.990908+00:00",  
      "created_by": "<username>"  
    }  
  ]  
}
```

6 Working with screenshots in Iris APIs

Retrieve the most recent screenshot for a given domain with Iris APIs in a two-step process:

1. [Retrieve the screenshot image URL](#)
2. [Retrieve and optionally resize the image file](#)

Also consult the section on [interpreting screenshot metadata](#).

6.1 Retrieve the screenshot URL

Append queries with `screenshots=1` to receive a screenshot JSON object in the response.

For example, the following Iris API queries (with header authentication) include screenshot requests.

6.1.1 Example queries for screenshot URL retrieval

- Iris Detect:
`https://api.domaintools.com/v1/iris-detect/domains/watched/?screenshots=1`
- Iris Enrich:
`https://api.domaintools.com/v1/iris-enrich/?domain=example.com&screenshots=1`
- Iris Investigate: `https://api.domaintools.com/v1/iris-investigate/?domain=example.com&screenshots=1`

6.1.2 Query parameters for screenshot URL retrieval

Query Parameter	Required	Type	Valid Values	Description	Example
<code>screenshots</code>	yes	flag	1	Triggers screenshot object in response.	<code>screenshots=1</code>

6.1.3 Response fields for screenshot URL retrieval

Output field	Type	Valid Values	Description	Example
<code>ip_address</code>	string null	IP address	The IP address from which the screenshot was taken.	212.129.31.169
<code>last_attempt</code>	string	ISO 8601 date-time format	The date and time the screenshot was last attempted.	2025-02-17T15:34:55Z
<code>last_seen</code>	string	ISO 8601 date-time format	The date and time the screenshot was last observed	2025-02-17T15:34:55Z
<code>screenshot</code>	object	JSON object	Details about the screenshot.	

Output field	Type	Valid Values	Description	Example
src	str	URL	Link to the screenshot.	https://screenshots.ne.domaintool image?s3v=1&rurl=http%3A%2F%2Fesp 03- 28T21%3A48%3A37Z&token=dec12499e0
timestamp	string	ISO 8601 date-time format	The date and time the provided screenshot (in fullsize field) was obtained.	2024-08- 05T16:17:45Z

6.1.4 Response example for screenshot URL retrieval

```
"screenshot": {
  "timestamp": "2025-03-07T19:53:44Z",
  "src": "https://screenshots.ar.domaintools.com/screenshot_im-
↪ age?s3v=1&rurl=http%3A%2F%2Fdomaintools.com&ts=2025-03-
↪ 07T19%3A53%3A44Z&token=2d6269e40f4bc48908b17d218e5647fc90c0034618e39ec20806910d3909785c",
  "ip_address": "141.193.213.21",
  "last_attempt": "2025-04-07T18:00:01Z",
  "last_seen": "2025-04-07T18:00:01Z"
}
```

6.2 Retrieve and optionally resize the screenshot image via its URL

Use the `src` URL in the returned screenshot object to download the screenshot image. The URL already contains the required parameters `domain`, `ts`, `rurl`, and `token`. Optionally add `resize_width` and `crop_height` to resize the image.

6.2.1 Query parameters for screenshot image file

Query Parameter	Required	Type	Valid Values	Description	Example
<code>resize_width</code>	no	integer	Width in pixels	If the value is less than the width of the image, the image will be shrunk horizontally to the given <code>resize_width</code> , preserving the aspect ratio.	<code>resize_width=500</code>

Query Parameter	Required	Type	Valid Values	Description	Example
crop_height	no	integer	Height in pixels	If the value is less than the height of the image, the image will be cropped vertically to the given crop_height, removing pixels from the bottom of the image.	crop_height=500

6.2.2 Example query for screenshot image file

```
api.domaintools.com/screenshot_
image?domain=domaintools.com&ts=2025-04-30T14:32:10Z&resize_width=500&crop_
height=500&rurl=https://www.domaintools.com/
```

6.3 Interpreting screenshot metadata

Use the `last_attempt`, `last_seen`, and `timestamp` to interpret the screenshot.

To confirm that the most recent screenshot was gathered during the current domain lifecycle—and isn't from an older, historic screenshot—compare the screenshot's `timestamp` with the domain's `first_seen` value in the API response. If the `timestamp` is later than `first_seen`, the screenshot was taken during the current domain lifecycle.

6.3.1 Initial capture

The screenshot was successfully captured during the first and only attempt. Since there have been no subsequent checks, all `timestamp` fields are identical.

Relationships: `timestamp = last_seen = last_attempt`

Response example:

```

"screenshot": {
  "timestamp": "2025-05-12T14:38:19Z",
  "src": "https://screenshots.ar.domaintools.com/screenshot_im-
↪ age?s3v=1&rurl=http%3A%2F%2Fxi.mi.fr&ts=2025-05-
↪ 12T14%3A38%3A19Z&token=5fbd3aa604682d26d9f1f69810a87bfd75ea25e56eadb3553cefcff2ba804f28",
  "ip_address": "76.76.21.21",
  "last_attempt": "2025-05-12T14:38:19Z",
  "last_seen": "2025-05-12T14:38:19Z"
}
```

6.3.2 Gather attempt was duplicate of initial screenshot

DomainTools re-checked a domain after previously taking an initial screenshot. The result was identical to the existing image and no new screenshot was generated.

Relationships: `timestamp < last_seen = last_attempt`

- `timestamp` remains unchanged, preserving the original capture time.
- `last_attempt` is updated to reflect the most recent screenshot check.
- `last_seen` is updated to indicate the most recent time the image was confirmed valid.
- Because no new image was needed, `last_seen` and `last_attempt` are equal.

Response example:

```
    "screenshot": {
      "timestamp": "2025-05-12T10:10:49Z",
      "src": "https://screenshots.ar.domaintools.com/screenshot_im-
↵ age?s3v=1&rurl=http%3A%2F%2Fdomaintools.co.in&ts=2025-05-
↵ 12T10%3A10%3A49Z&token=28d255010a09d1cc8a3326e1c1406971910293f09ddac0159eb35391f705be57",
      "ip_address": "104.219.250.192",
      "last_attempt": "2025-05-13T17:16:34Z",
      "last_seen": "2025-05-13T17:16:34Z"
    }
```

6.3.3 Recapture failed after previous successful capture

DomainTools successfully captured a screenshot on a previous attempt. The next attempt to capture the screenshot failed — the system was unable to retrieve a new image and could not confirm whether the content had changed.

Relationships: `timestamp = last_seen`, and `last_seen < last_attempt`

- `timestamp` reflects the time the screenshot was originally captured.
- `last_seen` is equal to `timestamp`, because that was the last time a screenshot was successfully observed
- `last_attempt` is more recent than `last_seen`, but the attempt failed

Response example:

```
    "screenshot": {
      "timestamp": "2023-12-19T22:17:10Z",
      "src": "https://screenshots.ar.domaintools.com/screenshot_im-
↵ age?s3v=1&rurl=http%3A%2F%2Ffixhelpdesk.com&ts=2023-12-
↵ 19T22%3A17%3A10Z&token=8cd6edb822594a0aa485e72f5cbb5414a73b657f7438b770844f46fe886c4ea9",
      "ip_address": "91.195.240.19",
      "last_attempt": "2025-05-13T10:25:04Z",
      "last_seen": "2023-12-19T22:17:10Z"
    }
```

6.3.4 Duplicate capture followed by failed attempt

A screenshot was captured, and subsequently re-captured as a duplicate. A subsequent attempt to refresh the (duplicate) screenshot failed.

Relationships: `timestamp < last_seen`, and `last_seen < last_attempt`

- `timestamp` reflects the original screenshot date
- `last_seen` is later than `timestamp`, indicating the image was revalidated as a duplicate on a later date
- `last_attempt` reflects a more recent failed attempt to recapture the screenshot

Example response:

```
    "screenshot": {
      "timestamp": "2022-07-02T10:03:43Z",
      "src": "https://screenshots.ar.domaintools.com/screenshot_image?
↵ s3v=1&rurl=http%3A%2F%2Fdomaintools.ca&ts=2022-07-
↵ 02T10%3A03%3A43Z&token=6fa06c07ff1b7206ae8f38aadaf4e11b8cf538b7625944e2813a72ebec2c6648",
      "ip_address": "158.85.87.76",
      "last_attempt": "2025-05-13T10:12:04Z",
      "last_seen": "2024-05-22T10:11:09Z"
    }
```